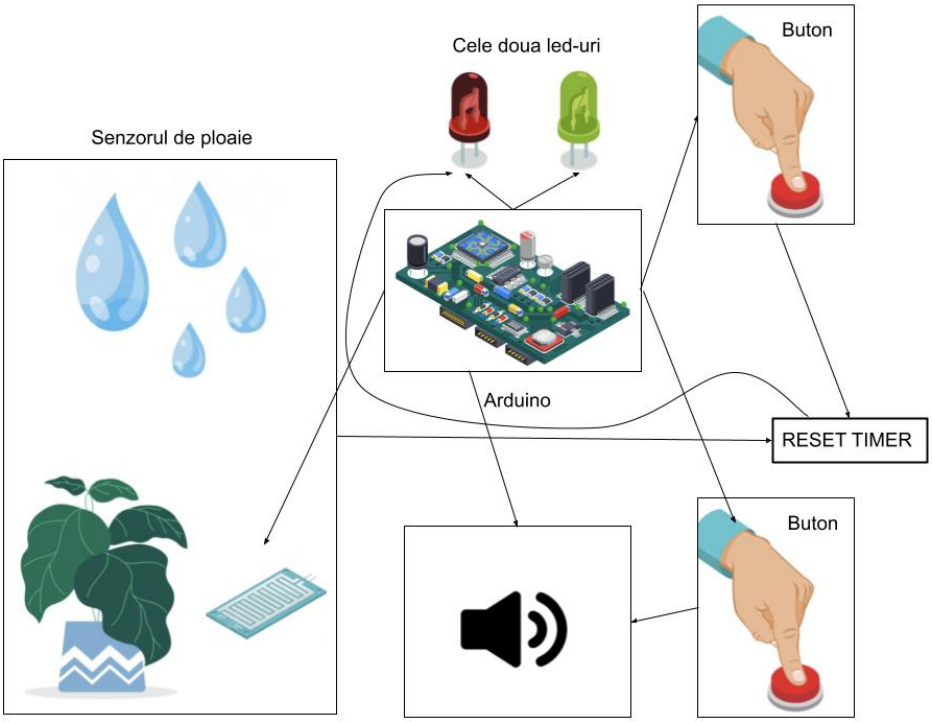
Sistem de monitorizare irigații plante - Ion Duiu

Introducere

- Scopul proiectului este de a realiza un sistem care îți aduce aminte când trebuie sa uzi plantele tale ținând cont de precipitațiile din natură.
- Am decis sa fac acest proiect deoarece foarte multe persoane cumpără sau primesc plante și uită să le îngrijească corespunzător, iar plantele ajung să ofilească deoarece nu sunt irigate periodic.

Descriere generală

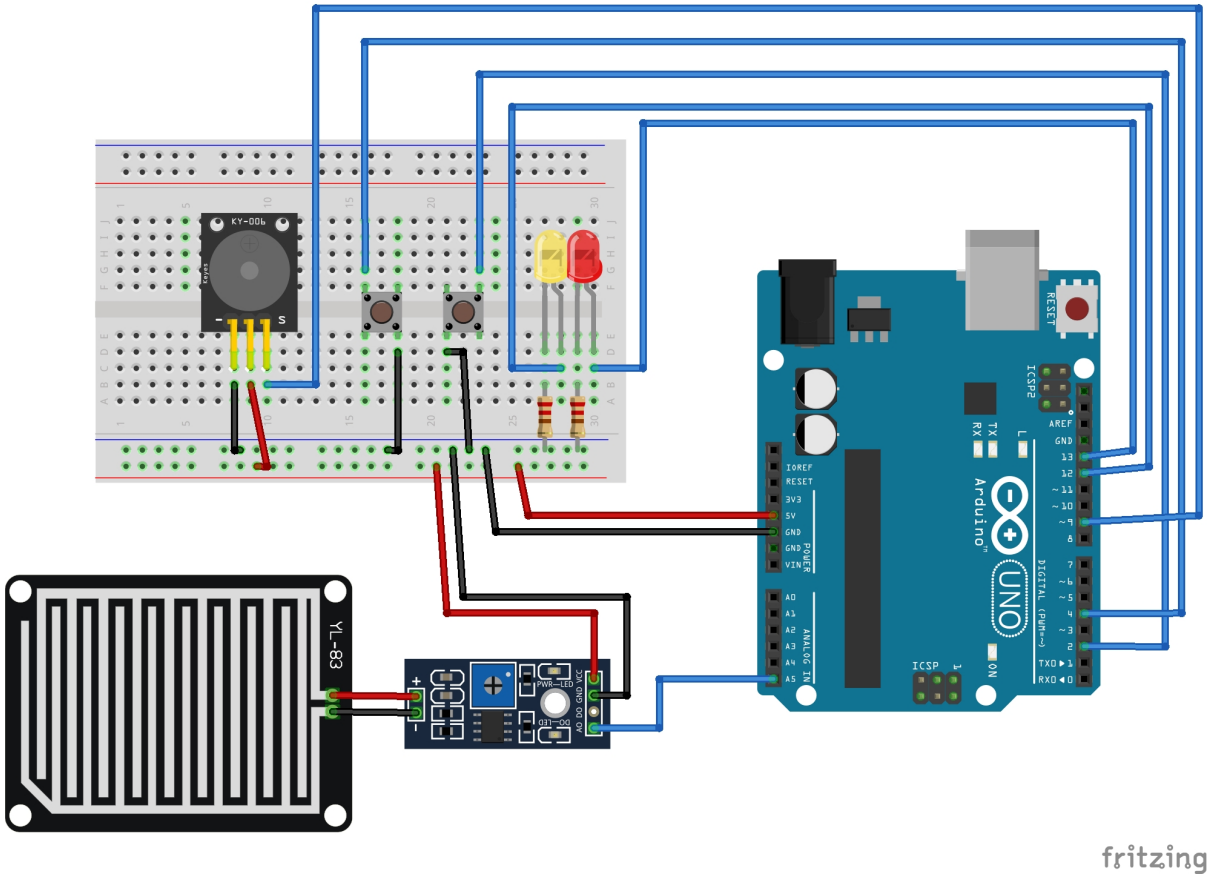
Proiectul are ca scop realizarea unui sistem ce ajută utilizatorul să aibă grijă de plantele sale. Utilizatorul va trebui sa urmărească cele doua leduri care ii vor spune dacă plantele sale au nevoie de apă sau nu. Când plantele au suficientă apă va fi aprins un led galben, iar când plantele au nevoie de apă se va aprinde un led roșu. Aceasta tranziție se face pe baza unui timer predefinit, dar pe langa acest timer, utilizatorul are la dispoziție un buton, care are ca scop resetarea timer-ului în cazul în care a decis sa ude plantele sale înainte de timer. Acest buton este util în cazul în care plantele nu au încă nevoie de apa, dar utilizatorul va fi plecat o perioada de timp și dorește sa ude preventiv plantele. Astfel în momentul în care se va întoarce, poate vedea dacă plantele lui au nevoie de apa sau au suficientă. Pe lângă acest buton va exista și un senzor de ploaie care va reseta timer-ul în momentul în care afara ploua și plantele au fost udate natural. Am decis sa adaug si un semnal acustic printr-un buzzer activ care poate fi pornit prin apasarea celui de-al doilea buton, astfel in momentul in care plantele trebuie irigate utilizatorul este anuntat si printr-un semnal acustic. Acest adaugare a fost gandita pentru a face dispozitivul folosibil si pentru cei care au probleme cu vederea.



Hardware Design

Piesele necesare:

- Placuta Arduino
- Cabluri
- Placuta de testare
- Senzor de ploaie
- Activ Buzzer
- Butoane
- LED-uri
- Rezistente



Software Design

Mediul de implementare ales a fost Arduino IDE.

Decizii de implementare:

- Timer-ul a fost implementat folosind Timer1, pe modul normal. Am calculat valoarea pentru TCNT1 ca fiind 3036 pentru ca timerul sa faca overflow la un interval de 1 secunda. Set-up-ul timer-ului are loc in functia setup de mai jos. In momentul in care se intampla overflow-ul, apelez functia onTimer si setez TCNT1 ca fiind 3036 din noua. In functia onTimer incrementez numarul de secunde care au trecut printr-o variabila deoarece aceasta functie se apeleaza la o secunda de fiecare data. Apoi in functie de valoarea variabilei *seconds* setez cele 2 led-uri si buzzer-ul daca acesta a fost activa de catre butonul care are ca functionalitate activarea sau dezactivarea buzzer-ului. De asemenea timer-ul este folosit pentru a supraveghea intervalul la care plantele trebuie irigate. Pentru acestea demonstratie valorea este setata la 15 secunde dar in practica valoarea ar trebui sa fie setata in functie de plantele pentru care este folosit sistemul.

```
void onTimer() {
  seconds++;
  if (seconds >= 15) {
    PORTB |= (1 << PB4);
    PORTB &= ~(1 << PB5);
    if(counter % 2 == 0) {
      PORTB |= (1 << PB1);
    } else {
      PORTB &= ~(1 << PB1);
```

```
    }
  } else {
    PORTB |= (1 << PB5);
    PORTB &= ~(1 << PB4);
    PORTB &= ~(1 << PB1);
  }
}

void setup() {
  DDRB |= (1 << PB1);
  DDRB |= (1 << PB4);
  DDRB |= (1 << PB5);
  DDRD &= ~(1 << PD2);
  PORTD |= (1 << PD2);
  DDRD &= ~(1 << PD4);
  PORTD |= (1 << PD4);
  cli();
  TCCR1A = 0;
  TCCR1B = 0;
  TCNT1 = 3036;
  TCCR1B |= (1 << CS12);
  TIMSK1 = (1 << TOIE1);
  PCMSK2 |= (1 << PCINT20);
  PCICR |= (1 << PCIE2);
  sei();
}

ISR(TIMER1_OVF_vect) {
  onTimer();
  TCNT1 = 3036;
}
```

- Butonul pentru activarea sau dezactivarea buzzer-ului foloseste intreruperi pentru a putea fi apasat oricand si a nu intrerupe flow-ului programului. Acesta foloseste variabila *counter* pentru a cunoaste starea buzzer-ului, astfel cand programul porneste *counter* este 0 si se incrementeaza la fiecare apasare de buton. Daca variabila *counter* are o valorea para atunci buzzerul nu trebuie sa porneasca in momentul in care plantele au nevoie de apa, iar cand valorea este impara buzzerul se va activa in acelasi moment in care se aprinde led-ul rosu pentru a indica si printr-un semnal acustic ca plantele au nevoie de apa. De asemenea in intrerupere pentru a ni se confirma alegerea facuta buzzer-ul este pornit pentru o perioada foarta scurta in momentul in care butonul este apasat pentru a activa buzzerul.

```
ISR(PCINT2_vect) {
  if (millis() - lastDebounceTime > debounceDelay) {
    if (buttonState == HIGH && digitalRead(4) == LOW) {
      counter++;
    }
    if(counter % 2 == 0) {
      PORTB |= (1 << PB1);
    } else {
      PORTB &= ~(1 << PB1);
```

```
    }
    buttonState = digitalRead(4);
    lastDebounceTime = millis();
  }
}
```

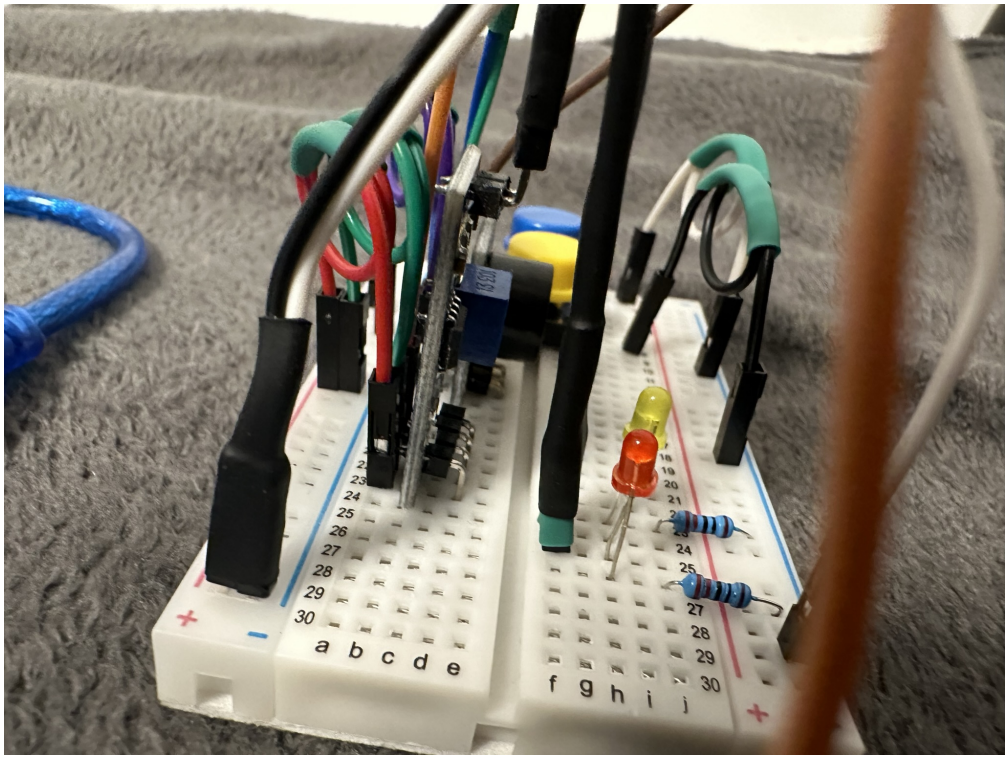
- Pentru senzorul de ploaie am folosit functia specifica de ADC din arduino si anume analogRead. Am testat si valoarea optima la care pe care ar trebui sa o returneze senzorul in momentul unei ploi este cuprinsa intre 250 si 350 dar pentru a putea demonstra cum functioneaza sistemul am lasat valoarea de 900. Astfel in momentul in care un deget atinge placa de detectare valorea curentului care trece prin placa se schimba datorita scaderii rezistentei si timer-ul se reseteaza. Acesta implementare, impreuna cu implementarea butonului care reseteaza manual timer-ul in momentul in care am irigat plantele se afla in functia loop de mai jos.

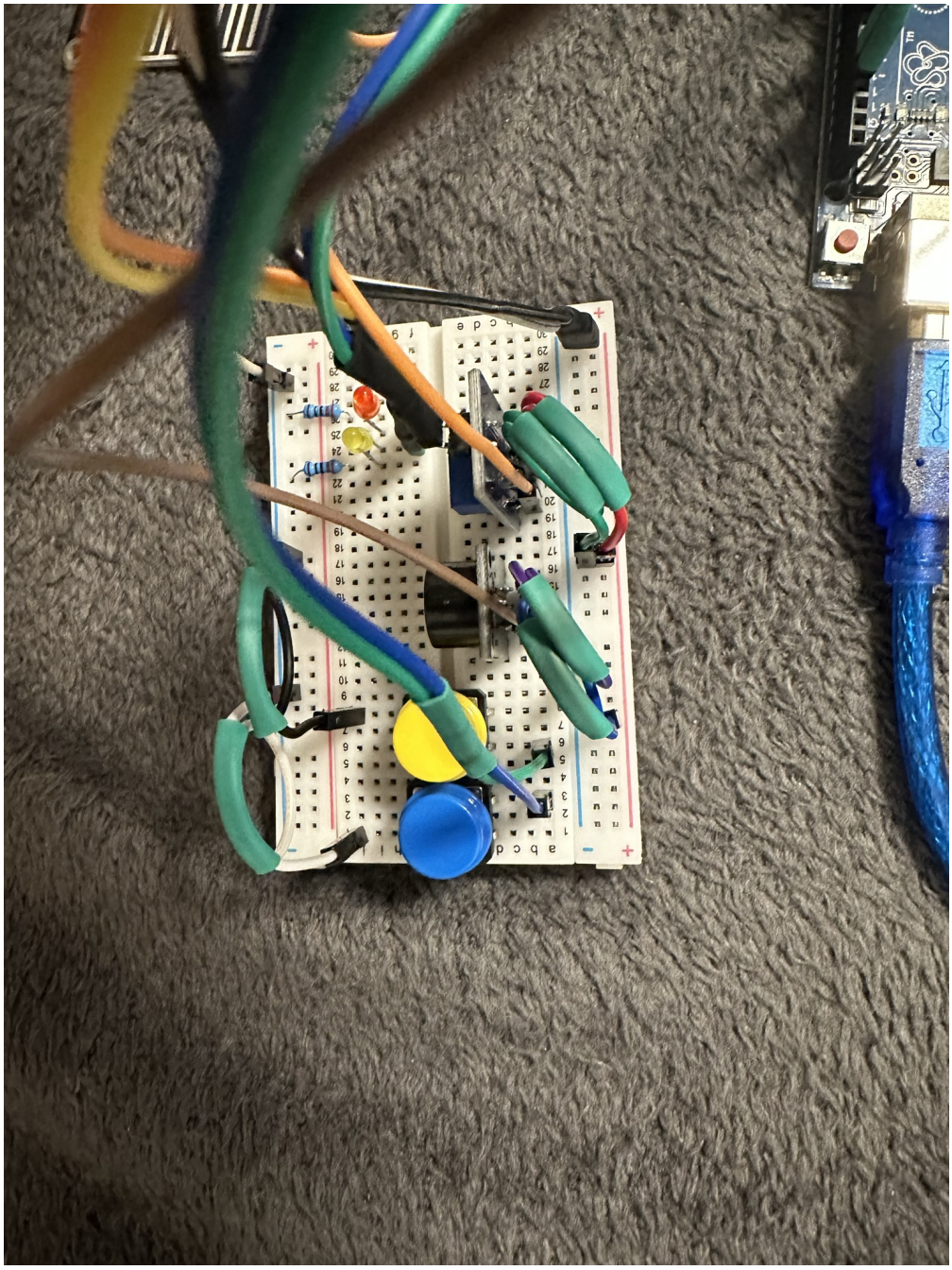
```
void loop() {
  int digitalVal = PIND & (1 << PD2);
  int sensorVal = analogRead(A5);

  if(digitalVal == LOW || sensorVal < 900) {
    seconds = 0;
  }

}
```

Rezultate Obţinute





Concluzii

Concluzii pentru proiectul dat:

1. Prin realizarea acestui proiect, se urmărește rezolvarea problemei udarii necorespunzătoare a plantelor, care duce la ofilirea acestora. Sistemul propus ajută utilizatorul să aibă grijă de plantele sale, prin indicarea momentului în care plantele au nevoie de apă.
2. Implementarea sistemului se bazează pe utilizarea unui timer predefinit, care indică momentul în care plantele trebuie udate. Utilizatorul poate reseta manual timer-ul prin intermediul unui buton, în cazul în care dorește să ude preventiv plantele sau în cazul în care va fi plecat o perioadă de timp.
3. Pentru a asigura precizia sistemului, s-a adăugat și un senzor de ploaie, care resetează timer-ul în momentul în care plantele sunt udate natural prin precipitații.

4. Pentru a oferi o notificare suplimentară utilizatorului, s-a implementat un semnal acustic prin intermediul unui buzzer activ. Acesta poate fi pornit manual prin apăsarea unui buton și servește drept avertizare sonoră în momentul în care plantele necesită udare.
5. Proiectul utilizează placuta Arduino și o serie de componente hardware, precum cabluri, senzor de ploaie, activ buzzer, butoane, LED-uri și rezistențe. Implementarea software se realizează în mediul de implementare Arduino IDE, iar deciziile de implementare includ utilizarea timer-ului Timer1 pentru gestionarea intervalului de timp și a intreruperilor pentru funcționalități specifice butoanelor.

În ansamblu, acest proiect propune o soluție utilă și practică pentru a ajuta utilizatorii să îngrijească plantele în mod corespunzător, evitând astfel ofilirea acestora ca urmare a udării insuficiente.

Download

duiuion332capm.zip

Bibliografie/Resurse

- Senzor Ploaie
- Laboratoarele de PM

Export to PDF